

---

# LABORATÓRIO 5

---

## CRIAÇÃO DE FUNÇÕES

### EXERCÍCIOS DE REVISÃO

---

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Existem muitas formas de escrever um programa com funções. Uma dúvida comum é o que colocar na função e o que deixar no programa principal. Veja, por exemplo, duas soluções diferentes para um mesmo programa.

```
// versão 1
#include <iostream>
using namespace std;

void alarme();
int lerSenha();

int main()
{
    cout << "Iniciando...\n";
    alarme();
    cout << "Senha: ";
    int senha = lerSenha();
}

void alarme()
{
    cout << '\a';
}

int lerSenha()
{
    int n;
    cin >> n;
    return n;
}
```

```
// versão 2
#include <iostream>
using namespace std;

void alarme();
void lerSenha();

int main()
{
    alarme();
    lerSenha();
}

void alarme()
{
    cout << "Iniciando...\n";
    cout << '\a';
}

void lerSenha()
{
    cout << "Senha: ";
    int senha;
    cin >> senha;
}
```

As duas versões geram o mesmo resultado para o usuário. Porém, as funções da 1ª versão estão orientadas a tarefas enquanto as funções da 2ª versão estão orientadas a blocos. **As duas formas são úteis e tem suas aplicações.**

Mas imagine que, após a leitura da senha, fosse necessário mostrar a mensagem "Finalizando..." e emitir outra vez o alarme. Como isso poderia ser feito em cada versão?

2. Variáveis globais podem ser acessadas e modificadas por qualquer função do programa. Justamente por essa razão elas devem ser evitadas. As variáveis globais geram programas difíceis de entender e fáceis de quebrar.

Para entender melhor, analise o programa abaixo.

```
#include <iostream>
using namespace std;

int num;

int f();
int g();

int main()
{
    num = 1;
    cout << f() + g() + num << endl;
}

int f()
{
    num = num + 3;
    return num;
}

int g()
{
    num = 2;
    return num;
}
```

- a. Qual é a ordem de chamada das funções dentro da função principal? Ou seja, quem será chamada primeiro, `f()` ou `g()`?

**Dica:** use o depurador para descobrir.

- b. Qual é o resultado do programa? Ele seria o mesmo se invertêssemos a ordem das chamadas de `f()` e `g()` dentro da função principal?

```
int main()
{
    num = 1;
    cout << g() + f() + num << endl;
}
```

- c. Existe alguma garantia de que as funções `f()` e `g()` serão sempre chamadas na ordem em que elas aparecem no programa?

## EXERCÍCIOS DE FIXAÇÃO

---

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Escreva um programa que peça ao usuário para entrar com um valor para hora e outro valor para minuto. A função `main()` deve então passar estes dois valores para uma função que mostra os valores na tela no formato hh:mm.

```
Entre com o número de horas: 9
Entre com o número de minutos: 45
Agora são 9:45
```

2. Escreva um programa para converter temperaturas de Celsius para Fahrenheit. Você deve criar uma função que receba um valor em Celsius e retorne o equivalente em Fahrenheit. O programa deve pedir ao usuário para digitar uma temperatura em graus Celsius, usar a função para obter o valor equivalente em graus Fahrenheit e exibir uma mensagem com o resultado. A saída do programa deve seguir o modelo abaixo:

```
Digite uma temperatura em graus Celsius: 20.5
20.5 graus Celsius equivalem a 68.9 graus Fahrenheit.
```

Fahrenheit =  $1.8 \times \text{Celsius} + 32.0$

3. Escreva um programa que peça ao usuário para entrar com dois números. O programa deve usar uma função para calcular a média harmônica dos números e retornar esse resultado para a função principal, que deve então apresentar o resultado na tela.

```
Entre com um número: 4.4
Entre com outro número: 6.2
A média harmônica dos números é 5.14717
```

A média harmônica de dois números  $x$  e  $y$  é igual a  $2.0 * x * y / (x+y)$ .

## EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Qual é a diferença entre as instruções de inicialização a seguir? Existe alguma razão para preferir uma opção sobre a outra? O resultado delas depende de onde elas estão dentro do código?

```
int x;  
x = 1;
```

```
int y = 1;
```

2. Defina uma função que aumenta o salário de um funcionário em 15%. Na função principal peça o salário atual do funcionário, passe este salário para uma função que retorna o salário aumentado. Na função principal mostre o novo salário do funcionário.

```
Salário atual: R$10000  
Salário ajustado para R$11500
```

3. Construa uma função que calcule o índice de massa corporal (IMC) de um indivíduo. A função receberá a altura (m) e a massa (kg) do indivíduo e retornará o seu IMC para a função principal. A função principal deve mostrar o resultado.

Use a fórmula:  $IMC = massa / (altura)^2$

```
Índice de Massa Corporal (IMC)  
-----  
Altura: 1.78  
Massa: 75  
IMC: 23.6713
```

4. A norma ou módulo de um vetor é um número real que representa o comprimento desse vetor. Dessa forma, calcular a norma de um vetor é o mesmo que calcular a distância entre um ponto (x,y) e a origem (0,0). Utilizando  $|v|$  como a notação para o módulo de um vetor  $v = (x,y)$  pertencente ao plano, teremos:

$$|v| = \sqrt{x^2 + y^2}$$

Crie uma função que receba as coordenadas (x,y) de um vetor e retorne seu módulo. Inclua a biblioteca `cmath` e utilize as funções `sqrt` e `pow` para calcular raiz quadrada e potenciação.

```
Digite as coordenadas do vetor:  
x: 10  
y: 5  
  
O tamanho do vetor é 11.1803
```

5. As coordenadas polares representam um vetor pelo seu ângulo com o eixo horizontal do plano cartesiano e pelo seu comprimento, também chamado de norma ou módulo. O ângulo que o vetor forma com a horizontal pode ser encontrado pelo arco tangente:

$$\theta = \text{atan2}\left(\frac{y}{x}\right)$$

A função `atan2` da biblioteca `<cmath>` retorna o ângulo em radianos. Para obter o ângulo em graus é preciso fazer a conversão, multiplicando o resultado por 180 e dividindo por  $\pi$ .

Escreva uma função que receba as coordenadas (x,y) de um vetor e retorne seu ângulo em graus.

```
Digite as coordenadas do vetor:  
x: 10  
y: 5  
  
O ângulo do vetor é 26.5651 graus.
```

6. Agora construa um programa que utilize as funções das questões 4 e 5 para ler as coordenadas de um vetor e mostrar suas coordenadas polares, isto é, seu comprimento e ângulo.

```
Digite as coordenadas do vetor:  
x: 10  
y: 5  
  
Coordenadas polares do vetor:  
(11.1803, 26.5651)
```

Quando o programa estiver funcionando, **separe-o em 3 arquivos**: um arquivo contendo apenas a função `main()`, um arquivo de inclusão contendo os protótipos das funções e um arquivo fonte contendo as definições das funções.

7. Crie 2 funções, uma que eleva um número ao quadrado, outra ao cubo. Na função principal, receba um valor do usuário e mostre o número recebido ao quadrado, ao cubo e, por fim, mostre o cubo do quadrado desse número. Ou seja, passe o resultado da função quadrado para a função cubo. Isso pode ser feito passando a chamada da função quadrado como argumento da função cubo.

```
Digite um valor: 2.1  
  
Quadrado = 4.41  
Cubo = 9.261  
Cubo do quadrado = 85.7661
```